

What is claimed is:

1. A virtual machine that executes a virtual machine instruction sequence under control of a real machine, comprising:

stack means for temporarily storing data in a last-in first-out format;

instruction storing means for storing the virtual machine instruction sequence and a plurality of sets of succeeding instruction information, wherein each virtual machine instruction in the virtual machine instruction sequence is associated with a set of succeeding instruction information that indicates a change in a storage state of the data in the stack means due to execution of a virtual machine instruction executed after the associated virtual machine instruction;

read means for reading a virtual machine instruction and an associated set of succeeding instruction information from the instruction storing means; and

decoding-executing means for specifying and executing operations corresponding to a combination of the read virtual machine instruction and the read set of succeeding instruction information.

2. The virtual machine of Claim 1, wherein the decoding-executing means includes:

a real machine instruction sequence storing unit for

4 storing a plurality of real machine instruction sequences  
5 that correspond to all combinations of virtual machine  
6 instructions and sets of succeeding instruction information;

7 a specifying unit for specifying a real machine  
8 instruction sequence in the real machine instruction  
9 sequence storing unit, the real machine instruction sequence  
10 corresponding to a combination of the virtual machine  
11 instruction and the set of succeeding instruction  
12 information read by the read means; and

13 an executing unit for executing the specified real  
14 machine instruction sequence.

1 3. The virtual machine of Claim 2, wherein each set of  
2 succeeding instruction information indicates a change in a  
3 number of sets of data in the stack means due to execution  
4 of a virtual machine instruction executed after a virtual  
5 machine instruction associated with the set of succeeding  
6 instruction information and

7 wherein at least one real machine instruction  
8 sequence stored in the real machine instruction sequence  
9 storing unit contains real machine instructions that perform  
10 a stack handling in the stack means in advance for a virtual  
11 machine instruction that is to be executed based on a set of  
12 succeeding instruction information associated with a  
13 currently executed virtual machine instruction.

1 4. The virtual machine of Claim 3, wherein the real  
2 machine instruction sequences stored in the real machine  
3 instruction sequence storing unit are composed with a  
4 premise that regions of the stack means used to store two  
5 sets of data to be read first and second are mapped to two  
6 registers in the real machine.

1 5. The virtual machine of Claim 1, wherein the instruction  
2 storing means includes a first storage area for storing the  
3 virtual machine instruction sequence and a second storage  
4 area for storing the sets of succeeding instruction  
5 information, wherein each location that stores a virtual  
6 machine instruction in the first storage area is associated  
7 with a location that stores an associated set of succeeding  
8 instruction information in the second storage area and  
9 wherein the read means reads the virtual machine  
10 instruction from a location in the first storage area and  
11 the associated set of succeeding instruction information  
12 from a location in the second storage area, the location in  
13 the first storage area being associated with the location in  
14 the second storage area.

1 6. The virtual machine of Claim 1, wherein the virtual  
2 machine instruction sequence stored in the instruction  
3 storing means is an extended virtual machine instruction  
4 sequence that includes extended virtual machine

5 instructions, the extended virtual machine instructions  
6 being combinations of virtual machine instructions and  
7 associated sets of succeeding instruction information,  
8 wherein the read means reads an extended virtual  
9 machine instruction from the instruction storing means, and  
10 wherein the decoding-executing means specifies and  
11 executes operations corresponding to the extended virtual  
12 machine instruction.

1 7. A compiler that generates programs for a virtual  
2 machine with a stack architecture that includes a stack,  
3 comprising:  
4 instruction sequence converting means for converting  
5 a source program into a virtual machine instruction sequence  
6 executable by the virtual machine;  
7 succeeding instruction information generating means  
8 for generating sets of succeeding instruction information  
9 corresponding to virtual machine instructions in the virtual  
10 machine instruction sequence, each set of succeeding  
11 instruction information indicating a change in a storage  
12 state of data in the stack due to execution of a virtual  
13 machine instruction executed after a virtual machine  
14 instruction corresponding to the set of succeeding  
15 instruction information; and  
16 associating means for associating each set of  
17 generated succeeding instruction information with a

18 corresponding virtual machine instruction and outputting the  
19 set of succeeding instruction information and the virtual  
20 machine instruction.

1 8. A virtual machine that executes a virtual machine  
2 instruction sequence under control of a real machine,  
3 comprising:

4 instruction storing means for storing the virtual  
5 machine instruction sequence;

6 read means for reading a virtual machine instruction  
7 in the virtual machine instruction sequence from the  
8 instruction storing means; and

9 decoding-executing means for specifying and executing  
10 operations corresponding to the virtual machine instruction,

11 wherein the decoding-executing means includes

12 a branch instruction judging unit for judging if the  
13 virtual machine instruction is a branch instruction and

14 an interrupt handling unit for detecting, if the  
15 virtual machine instruction is judged to be a branch  
16 instruction, whether there is an interrupt request, and, if  
17 so, performing a corresponding interrupt handling in  
18 addition to executing the branch instruction.

1 9. The virtual machine of Claim 8, wherein the decoding-  
2 executing means further includes

3 a real machine instruction sequence storing unit for

4 storing real machine instruction sequences corresponding to  
5 every virtual machine instruction and real machine  
6 instruction sequences for having interrupt handling  
7 performed corresponding to each interrupt request and  
8 an executing unit for executing a real machine  
9 instruction sequence corresponding to the virtual machine  
10 instruction read by the read means,

11 wherein if the virtual machine instruction is judged  
12 to be the branch instruction and an interrupt request is  
13 detected, the interrupt handling unit has the executing unit  
14 execute a real machine instruction sequence for having the  
15 corresponding interrupt handling performed and then the real  
16 machine instruction sequence corresponding to the branch  
17 instruction.

10. A virtual machine that executes a virtual machine  
instruction sequence under control of a real machine,  
comprising:

4 instruction storing means for storing the virtual  
5 machine instruction sequence;

6 read means for reading a virtual machine instruction  
7 in the virtual machine instruction sequence from the  
8 instruction storing means; and

9 decoding-executing means for specifying and executing  
10 operations corresponding to the read virtual machine  
11 instruction,

12 wherein the decoding-executing means includes  
13 a block judging unit for judging if the read virtual  
14 machine instruction is a virtual machine instruction  
15 representative of a block, a block being a predetermined  
16 number of virtual machine instructions and  
17 an interrupt handling unit for detecting, if the read  
18 virtual machine instruction is judged to be the  
19 representative virtual machine instruction, whether there is  
20 an interrupt request to the virtual machine, and if so,  
21 performing a corresponding interrupt handling in addition to  
22 executing the representative virtual machine instruction.

1 11. The virtual machine of Claim 10, wherein the decoding-  
2 executing means includes

3 a real machine instruction sequence storing unit for  
4 storing a plurality of real machine instruction sequences  
5 corresponding to every virtual machine instruction and at  
6 least one real machine instruction sequence for having  
7 interrupt handling performed in response to an interrupt  
8 request and

9 an executing unit for executing a real machine  
10 instruction sequence corresponding to the read virtual  
11 machine instruction,

12 wherein the block judging unit judges that the read  
13 virtual machine instruction is a virtual machine instruction  
14 representative of the block when a number of virtual machine

15 instructions that have been read is equal to a multiple of  
16 the predetermined number and

17 wherein if the read virtual machine instruction is  
18 judged to be a representative virtual machine instruction  
19 and an interrupt request has been detected, the interrupt  
20 handling unit has the executing unit execute a real machine  
21 instruction sequence for having the interrupt handling  
22 performed and then the real machine instruction sequence  
23 corresponding to the representative virtual machine  
24 instruction.

1 12. A virtual machine that executes a virtual machine  
2 instruction sequence under control of a real machine,  
3 comprising:

4 real machine program storing means for storing a  
5 plurality of subprograms composed of real machine  
6 instructions;

7 instruction storing means that includes a first area  
8 for storing the virtual machine instruction sequence and a  
9 second area for storing a plurality of pointers to the  
10 subprograms in the real machine program storing means;

11 read means for reading a virtual machine instruction  
12 in the virtual machine instruction sequence from the first  
13 area in the instruction storing means; and

14 decoding-executing means for specifying and executing  
15 operations corresponding to the read virtual machine



16 instruction,

17 wherein the decoding-executing means includes  
18 an area judging unit for judging whether the virtual  
19 machine instruction is an instruction that transfers control  
20 flow to a location in the second area and  
21 an address converting-executing unit for executing,  
22 if the virtual machine instruction is judged to be an  
23 instruction that transfers control flow to a location in the  
24 second area, a subprogram indicated by a pointer stored in  
25 the location.

1 13. The virtual machine of Claim 12, wherein the first  
2 area and the second area in the instruction storing means  
3 are two adjacent storage areas whose boundary is marked by  
4 an address and

5 wherein the area judging unit judges, when the read  
6 virtual machine instruction is a call instruction for a  
7 subprogram, whether the virtual machine instruction is an  
8 instruction that transfers control flow, by comparing a call  
9 address of the call instruction with the address.

1 14. A virtual machine that executes a virtual machine  
2 instruction sequence under control of a real machine,  
3 comprising:

4 instruction storing means for storing the virtual  
5 machine instruction sequence;

6 read means for reading a virtual machine instruction  
7 in the virtual machine instruction sequence from the  
8 instruction storing means; and

9 decoding-executing means for specifying and executing  
10 operations corresponding to the read virtual machine  
11 instruction,

12 wherein the instruction storing means is a plurality  
13 of instruction blocks that constitute the virtual machine  
14 instruction sequence, the instruction blocks corresponding  
15 to basic blocks,

16 wherein the instruction blocks each include: an  
17 identifier area for storing an identifier that specifies a  
18 start position of the instruction block in the instruction  
19 storing means; a non-branch instruction area for storing  
20 non-branch instructions belonging to a corresponding basic  
21 block; and a branch instruction area for storing at least  
22 one branch instruction belonging to the corresponding basic  
23 block,

24 wherein each branch instruction stored in the branch  
25 instruction area designates a branch destination using an  
26 identifier stored in one of the identifier areas, and

27 wherein if the read virtual machine instruction is a  
28 branch instruction, the decoding-executing means has control  
29 flow branch to a start position of a non-branch instruction  
30 area in an instruction block having an identifier designated  
31 by the branch instruction as a branch destination.

1 15. The virtual machine of Claim 14, wherein the decoding-  
2 executing means includes a program counter composed of (a)  
3 an identifier register for storing an identifier of an  
4 instruction block to which a virtual machine instruction to  
5 be read belongs and (b) an offset counter for storing an  
6 offset that indicates a relative storage position of the  
7 virtual machine instruction in the instruction block,

8 wherein the read means reads the virtual machine  
9 instruction based on the identifier and the offset in the  
10 program counter,

11 wherein the decoding-executing means updates, if the  
12 read virtual machine instruction is the branch instruction,  
13 the program counter by writing the identifier designated as  
14 the branch destination by the branch instruction into the  
15 identifier register and by setting an initial value in the  
16 offset counter, and if the read virtual machine instruction  
17 is a non-branch instruction, updates the program counter by  
18 incrementing the offset counter, and

19 wherein the read means reads a virtual machine  
20 instruction to be executed next based on the program counter  
21 updated by the decoding-executing means.

1 16. The virtual machine of Claim 15, wherein the decoding-  
2 executing means includes a real machine instruction sequence  
3 storing unit that stores a plurality of real machine

4 instruction sequences that each correspond to a different  
5 virtual machine instruction,

6 wherein the instruction blocks in the instruction  
7 storing means each include a decoded data sequence area for  
8 storing a decoded data sequence that specifies real machine  
9 instruction sequences in the real machine instruction  
10 sequence storing unit, the real machine instruction  
11 sequences corresponding to virtual machine instructions  
12 stored in the non-branch instruction area and the branch  
13 instruction area of the instruction block,

14 wherein if a decoded data sequence is stored in an  
15 instruction block where reading is to be performed, the read  
16 means reads a set of decoded data in the decoded data  
17 sequence instead of a virtual machine instruction, and if  
18 not, the read means reads the virtual machine instruction  
19 and then generates a set of decoded data to specify a real  
20 machine instruction sequence in the real machine instruction  
21 sequence storing unit that corresponds to the virtual  
22 machine instruction, and

23 wherein the decoding-executing means reads from the  
24 real machine instruction sequence storing unit the real  
25 machine instruction sequence specified by the set of decoded  
26 data that has been either read or generated by the read  
27 means, and executes the real machine instruction sequence.

1 17. The virtual machine of Claim 16, wherein the decoded

2 data sequence area in the instruction storing means includes  
3 a flag area for storing a flag that indicates whether the  
4 decoded data sequence is stored in the decoded data sequence  
5 area,

6 wherein the decoding-executing means includes a  
7 current flag storing unit for storing a flag that is read  
8 from a flag area in a branch destination instruction block  
9 by the decoding-executing means when executing a branch  
10 instruction, and

11 wherein the read means reads a set of decoded data or  
12 a virtual machine instruction depending on the flag in the  
13 current flag storing unit.

1 18. The virtual machine of Claim 16, wherein each  
2 instruction block in the instruction storing means further  
3 includes a flag area for storing a flag that indicates  
4 whether a decoded data sequence is stored in the decoded  
5 data sequence area of the instruction block and

6 wherein the decoding-executing means includes a  
7 decoded data sequence writing unit for judging, after a  
8 branch instruction has been executed, whether the  
9 instruction block designated as the branch destination by  
10 the branch instruction stores a decoded data sequence by  
11 referring to a flag stored in a flag area of the instruction  
12 block, and if no decoded data sequence is stored, having a  
13 virtual machine instruction sequence in the instruction

14 block read, decoding the read virtual machine instruction  
15 sequence to produce a decoded data sequence, and writing the  
16 decoded data sequence into a decoded data sequence area in  
17 the instruction block.

1 19. A virtual machine that executes a virtual machine  
2 instruction sequence under control of a real machine,  
3 comprising:

4 instruction storing means for storing a compressed  
5 virtual machine instruction sequence to be executed;

6 read means for reading a compressed virtual machine  
7 instruction in the compressed virtual machine instruction  
8 sequence from the instruction storing means and  
9 decompressing the compressed virtual machine instruction to  
10 generate a decompressed virtual machine instruction; and

11 decoding-executing means for specifying and executing  
12 operations corresponding to the decompressed virtual machine  
13 instruction,

14 wherein the instruction storing means is a plurality  
15 of instruction blocks containing compressed virtual machine  
16 instructions constituting the compressed virtual machine  
17 instruction sequence, the instruction blocks corresponding  
18 to basic blocks,

19 wherein the instruction blocks each include: an  
20 identifier area for storing an identifier that specifies a  
21 start position of the instruction block in the instruction

22 storing means; a non-branch instruction area for storing  
23 compressed non-branch instructions belonging to a  
24 corresponding basic block; and a branch instruction area for  
25 storing at least one compressed branch instruction belonging  
26 to the corresponding basic block,

27 wherein each compressed branch instruction stored in  
28 a branch instruction area designates a branch destination  
29 using an identifier stored in one of the identifier areas,  
30 and

31 wherein if the decompressed virtual machine  
32 instruction is a branch instruction, the decoding-executing  
33 means has control flow branch to a start position of a non-  
34 branch instruction area in an instruction block having an  
35 identifier designated by the branch instruction as a branch  
36 destination.

1 20. The virtual machine of Claim 19, wherein each  
2 instruction block includes a decompression table area for  
3 storing a decompression table for use during decompression  
4 of compressed virtual machine instructions in the  
5 instruction block, the decompression table containing at  
6 least one combination of a compressed virtual machine  
7 instruction stored in the instruction block and a  
8 corresponding decompressed virtual machine instruction and  
9 wherein the read means reads the compressed virtual  
10 machine instruction from the instruction storing means and

11 decompresses the compressed virtual machine instruction by  
12 referring to a decompression table in an instruction block  
13 to which the compressed virtual machine instruction belongs  
14 to generate the decompressed virtual machine instruction.

1 21. The virtual machine of Claim 20, wherein the decoding-  
2 executing means includes a program counter composed of (a)  
3 an identifier register for storing an identifier of an  
4 instruction block to which a compressed virtual machine  
5 instruction to be read belongs and (b) an offset counter for  
6 storing an offset that indicates a relative storage position  
7 of the compressed virtual machine instruction in the  
8 instruction block,

9 wherein the read means reads the compressed virtual  
10 machine instruction based on the identifier and the offset  
11 in the program counter,

12 wherein if the decompressed virtual machine  
13 instruction is a branch instruction, the decoding-executing  
14 means updates the program counter by writing the identifier  
15 designated as the branch destination by the branch  
16 instruction into the identifier register and by setting an  
17 initial value in the offset counter, and if the decompressed  
18 virtual machine instruction is a non-branch instruction,  
19 updates the program counter by incrementing the offset  
20 counter, and

21 wherein the read means reads a compressed virtual



22 machine instruction to be executed next based on the program  
23 counter updated by the decoding-executing means.

1 22. The virtual machine of Claim 21, wherein the decoding-  
2 executing means includes a real machine instruction sequence  
3 storing unit that stores a plurality of real machine  
4 instruction sequences that each correspond to a different  
5 virtual machine instruction,

6 wherein the instruction blocks in the instruction  
7 storing means each include a decoded data sequence area for  
8 storing a decoded data sequence that specifies real machine  
9 instruction sequences in the real machine instruction  
10 sequence storing unit, the real machine instruction  
11 sequences corresponding to compressed virtual machine  
12 instructions stored in the non-branch instruction area and  
13 the branch instruction area in the instruction block,

14 wherein if a decoded data sequence is stored in an  
15 instruction block where reading is to be performed, the read  
16 means reads a set of decoded data in the decoded data  
17 sequence instead of a compressed virtual machine  
18 instruction, and if not, the read means reads a compressed  
19 virtual machine instruction, decompresses the compressed  
20 virtual machine instruction to generate a decompressed  
21 virtual machine instruction, and then generates a set of  
22 decoded data to specify a real machine instruction sequence  
23 corresponding to the decompressed virtual machine

24 instruction in the real machine instruction sequence storing  
25 unit, and

26 wherein the decoding-executing means reads from the  
27 real machine instruction sequence storing unit the real  
28 machine instruction sequence specified by a set of decoded  
29 data that has been either read or generated by the read  
30 means, and executes the real machine instruction sequence.

1 23. The virtual machine of Claim 22, wherein each  
2 instruction block in the instruction storing means further  
3 includes a flag area for storing a flag that indicates  
4 whether a decoded data sequence is stored in the decoded  
5 data sequence area of the instruction block,

6 wherein the decoding-executing means includes a  
7 current flag storing unit for storing a flag that is read  
8 from a flag area in a branch destination instruction block  
9 by the decoding-executing means when executing a branch  
10 instruction, and

11 wherein the read means reads a set of decoded data or  
12 a compressed virtual machine instruction depending on the  
13 flag in the current flag storing unit.

1 24. The virtual machine of Claim 22, wherein each  
2 instruction block in the instruction storing means further  
3 includes a flag area for storing a flag that indicates  
4 whether a decoded data sequence is stored in the decoded

5 data sequence area of the instruction block and  
6 wherein the decoding-executing means includes a  
7 decoded data sequence writing unit for judging, after a  
8 branch instruction has been executed, whether the  
9 instruction block designated as the branch destination by  
10 the branch instruction stores a decoded data sequence by  
11 referring to a flag stored in a flag area of the instruction  
12 block, and if no decoded data sequence is stored, having a  
13 compressed virtual machine instruction sequence in the  
14 instruction block read and decompressed, having the  
15 decompressed virtual machine instruction sequence decoded to  
16 produce a decoded data sequence, and writing the decoded  
17 data sequence into a decoded data sequence area in the  
18 instruction block.

1 25. A Just-In-Time (JIT) compiler for use with a virtual  
2 machine that executes a virtual machine instruction sequence  
3 under control of a real machine, the JIT compiler converting  
4 parts of the virtual machine instruction sequence into real  
5 machine instruction sequences before execution, and

6 the JIT compiler comprising:

7 block start information receiving means for receiving  
8 an input of block start information for each virtual machine  
9 instruction that composes the virtual machine instruction  
10 sequence, the block start information showing whether a  
11 corresponding virtual machine instruction would correspond

12 to a start of a basic block if the virtual machine  
13 instruction sequence were divided into basic blocks;  
14 converting means for converting virtual machine  
15 instructions in the virtual machine instruction sequence  
16 into real machine instruction sequences; and  
17 outputting means for rearranging the real machine  
18 instruction sequences produced by the converting means into  
19 basic block units in accordance with the block start  
20 information received by the block start information  
21 receiving means.

26. The JIT compiler of Claim 25, further comprising  
branch violation judging means for judging, when a real  
machine instruction at a start of a produced real machine  
instruction sequence corresponds to a virtual machine  
instruction whose block start information indicates that the  
virtual machine instruction would be a start of a basic  
block, whether the real machine instruction is going to be  
arranged in an address that violates an address alignment  
restriction of the real machine,

wherein if the real machine instruction is going to  
be arranged in an address that violates the address  
alignment restriction, the outputting means rearranges the  
real machine instruction sequence so that the real machine  
instruction is not arranged in the address.

1 27. The JIT compiler of Claim 26, wherein the outputting  
2 means rearranges the real machine instruction sequence by  
3 inserting a necessary number of no-operation instructions at  
4 the start of the basic block to which the real machine  
5 instruction belongs.

1 28. The JIT compiler of Claim 25, wherein the outputting  
2 means inserts a certain number of no-operation instructions  
3 at a start of each basic block, the number being a number of  
4 real machine instructions processed during a delay of a  
5 delayed branch.

1 29. A storage method used by instruction storing means  
2 that stores a virtual machine instruction sequence to be  
3 executed by a virtual machine, having a stack architecture,  
4 under control of a real machine,

5 the storage method being characterized by storing  
6 each virtual machine instruction in the virtual machine  
7 instruction sequence associated with different succeeding  
8 instruction information, the succeeding instruction  
9 information for a given virtual machine instruction  
10 indicating a change in a storage state of data in a stack  
11 due to execution of a virtual machine instruction executed  
12 after the given virtual machine instruction.

1 30. A storage method used by instruction storing means

2 that stores a virtual machine instruction sequence to be  
3 executed by a virtual machine under control of a real  
4 machine,

5 wherein the storage method results in:

6 the instruction storing means being a plurality of  
7 instruction blocks that constitute the virtual machine  
8 instruction sequence, the instruction blocks corresponding  
9 to basic blocks;

10 the instruction blocks each including:

11 an identifier area for storing an identifier  
12 that specifies a start position of the  
13 instruction block in the instruction storing  
14 means;

15 a non-branch instruction area for storing  
16 non-branch instructions belonging to a  
17 corresponding basic block; and  
18 a branch instruction area for storing at  
19 least one branch instruction belonging to the  
20 corresponding basic block; and

21 each branch instruction stored in the branch  
22 instruction area designating a branch destination using an  
23 identifier stored in one of the identifier areas.

1 31. A computer-readable recording medium that stores a  
2 program to have a computer function as a virtual machine  
3 with a stack architecture,

4 wherein the virtual machine comprises:  
5 stack means for temporarily storing data in a last-in  
6 first-out format;  
7 instruction storing means for storing a virtual  
8 machine instruction sequence and a plurality of sets of  
9 succeeding instruction information, wherein each virtual  
10 machine instruction in the virtual machine instruction  
11 sequence is associated with a set of succeeding instruction  
12 information that indicates a change in a storage state of  
13 the data in the stack means due to execution of a virtual  
14 machine instruction executed after the associated virtual  
15 machine instruction;  
16 read means for reading a virtual machine instruction  
17 and an associated set of succeeding instruction information  
18 from the instruction storing means; and  
19 decoding-executing means for specifying and executing  
20 operations corresponding to a combination of the read  
21 virtual machine instruction and the read set of succeeding  
22 instruction information.

1 32. A computer-readable recording medium that stores a  
2 program to have a computer function as a compiler that  
3 generates a program for a virtual machine with a stack  
4 architecture,

5 wherein the compiler comprises:  
6 instruction sequence converting means for converting

7 a source program into a virtual machine instruction sequence  
8 executable by the virtual machine;

9 succeeding instruction information generating means  
10 for generating sets of succeeding instruction information  
11 corresponding to virtual machine instructions in the virtual  
12 machine instruction sequence, each set of succeeding  
13 instruction information indicating a change in a storage  
14 state of data in the stack due to execution of a virtual  
15 machine instruction executed after a virtual machine  
16 instruction corresponding to the set of succeeding  
17 instruction information; and

18 associating means for associating each set of  
19 generated succeeding instruction information with a  
20 corresponding virtual machine instruction and outputting the  
21 set of succeeding instruction information and the virtual  
22 machine instruction.

1 33. A computer-readable recording medium that stores a  
2 program to have a computer function as a virtual machine,  
3 wherein the virtual machine comprises:

4 instruction storing means for storing a virtual  
5 machine instruction sequence;

6 read means for reading a virtual machine instruction  
7 in the virtual machine instruction sequence from the  
8 instruction storing means; and

9 decoding-executing means for specifying and executing



10 operations corresponding to the virtual machine instruction,  
11 wherein the decoding-executing means includes  
12 a branch instruction judging unit for judging if the  
13 virtual machine instruction is a branch instruction and  
14 an interrupt handling unit for detecting, if the  
15 virtual machine instruction is judged to be a branch  
16 instruction, whether there is an interrupt request, and, if  
17 so, performing a corresponding interrupt handling in  
18 addition to executing the branch instruction.

1 34. A computer-readable recording medium that stores a  
2 program to have a computer function as a virtual machine,  
3 wherein the virtual machine comprises:  
4 instruction storing means for storing a virtual  
5 machine instruction sequence;  
6 read means for reading a virtual machine instruction  
7 in the virtual machine instruction sequence from the  
8 instruction storing means; and  
9 decoding-executing means for specifying and executing  
10 operations corresponding to the read virtual machine  
11 instruction,  
12 wherein the decoding-executing means includes  
13 a block judging unit for judging if the read virtual  
14 machine instruction is a virtual machine instruction  
15 representative of a block, a block being a predetermined  
16 number of virtual machine instructions and

17 an interrupt handling unit for detecting, if the read  
18 virtual machine instruction is judged to be the  
19 representative virtual machine instruction, whether there is  
20 an interrupt request to the virtual machine, and if so,  
21 performing a corresponding interrupt handling in addition to  
22 executing the representative virtual machine instruction.

1 35. A computer-readable recording medium that stores a  
2 program to have a computer function as a virtual machine,  
3 wherein the virtual machine comprises:

4 real machine program storing means for storing a  
5 plurality of subprograms composed of real machine  
6 instructions;

7 instruction storing means that includes a first area  
8 for storing a virtual machine instruction sequence and a  
9 second area for storing a plurality of pointers to the  
10 subprograms in the real machine program storing means;

11 read means for reading a virtual machine instruction  
12 in the virtual machine instruction sequence from the first  
13 area in the instruction storing means; and

14 decoding-executing means for specifying and executing  
15 operations corresponding to the read virtual machine  
16 instruction,

17 wherein the decoding-executing means includes  
18 an area judging unit for judging whether the virtual  
19 machine instruction is an instruction that transfers control

20 flow to a location in the second area and  
21 an address converting-executing unit for executing,  
22 if the virtual machine instruction is judged to be an  
23 instruction that transfers control flow to a location in the  
24 second area, a subprogram indicated by a pointer stored in  
25 the location.

1 36. A computer-readable recording medium that stores a  
2 program to have a computer function as a virtual machine,  
3 wherein the virtual machine comprises:  
4 instruction storing means for storing a virtual  
5 machine instruction sequence;  
6 read means for reading a virtual machine instruction  
7 in the virtual machine instruction sequence from the  
8 instruction storing means; and  
9 decoding-executing means for specifying and executing  
10 operations corresponding to the read virtual machine  
11 instruction,  
12 wherein the instruction storing means is a plurality  
13 of instruction blocks that constitute the virtual machine  
14 instruction sequence, the instruction blocks corresponding  
15 to basic blocks,  
16 wherein the instruction blocks each include: an  
17 identifier area for storing an identifier that specifies a  
18 start position of the instruction block in the instruction  
19 storing means; a non-branch instruction area for storing

20 non-branch instructions belonging to a corresponding basic  
21 block; and a branch instruction area for storing at least  
22 one branch instruction belonging to the corresponding basic  
23 block,

24 wherein each branch instruction stored in the branch  
25 instruction area designates a branch destination using an  
26 identifier stored in one of the identifier areas, and

27 wherein if the read virtual machine instruction is a  
28 branch instruction, the decoding-executing means has control  
29 flow branch to a start position of a non-branch instruction  
30 area in an instruction block having an identifier designated  
31 by the branch instruction as a branch destination.

1 37. A computer-readable recording medium that stores a  
2 program to have a computer function as a virtual machine,

3 wherein the virtual machine comprises:

4 instruction storing means for storing a compressed  
5 virtual machine instruction sequence to be executed;

6 read means for reading a compressed virtual machine  
7 instruction in the compressed virtual machine instruction  
8 sequence from the instruction storing means and  
9 decompressing the compressed virtual machine instruction to  
10 generate a decompressed virtual machine instruction; and

11 decoding-executing means for specifying and executing  
12 operations corresponding to the decompressed virtual machine  
13 instruction,

14 wherein the instruction storing means is a plurality  
15 of instruction blocks containing compressed virtual machine  
16 instructions constituting the compressed virtual machine  
17 instruction sequence, the instruction blocks corresponding  
18 to basic blocks,

19 wherein the instruction blocks each include: an  
20 identifier area for storing an identifier that specifies a  
21 start position of the instruction block in the instruction  
22 storing means; a non-branch instruction area for storing  
23 compressed non-branch instructions belonging to a  
24 corresponding basic block; and a branch instruction area for  
25 storing at least one compressed branch instruction belonging  
26 to the corresponding basic block,

27 wherein each compressed branch instruction stored in  
28 a branch instruction area designates a branch destination  
29 using an identifier stored in one of the identifier areas,  
30 and

31 wherein if the decompressed virtual machine  
32 instruction is a branch instruction, the decoding-executing  
33 means has control flow branch to a start position of a non-  
34 branch instruction area in an instruction block having an  
35 identifier designated by the branch instruction as a branch  
36 destination.

1 38. A computer-readable recording medium that stores a  
2 program to have a computer function as a Just-In-Time (JIT)

3 compiler used with a virtual machine that executes a virtual  
4 machine instruction sequence under control of a real  
5 machine, the JIT compiler converting parts of the virtual  
6 machine instruction sequence into real machine instruction  
7 sequences before execution,

8 wherein the compiler comprises:

9 block start information receiving means for receiving  
10 an input of block start information for each virtual machine  
11 instruction that composes the virtual machine instruction  
12 sequence, the block start information showing whether a  
13 corresponding virtual machine instruction would correspond  
14 to a start of a basic block if the virtual machine  
15 instruction sequence were divided into basic blocks;

16 converting means for converting virtual machine  
17 instructions in the virtual machine instruction sequence  
18 into real machine instruction sequences; and

19 outputting means for rearranging the real machine  
20 instruction sequences produced by the converting means into  
21 basic block units in accordance with the block start  
22 information received by the block start information  
23 receiving means.